УДК 519.248:629.7.017

Oleksandr. M. Khoshaba[1]

INVESTIGATION OF THE COMPUTING PERFORMANCE SYSTEM BASED ON QUEUING NETWORKS

[1] Vinnitsia National Technical University

Анотація

У роботі розглянуто дослідження продуктивності обчислювальних систем на основі мереж масового обслуговування. Показано історію розвитку моделей, переваги та недоліки їх роботи. Основна увага приділена програмному інструменту qnetworks, який широко використовується в даний час.

Ключові слова: продуктивність обчислювальних систем, мережі масового обслуговування, моделі, пакети прикладних програм, програмні інструменти.

Abstract

The article studies the computing systems' performance based on queuing networks. The focus is on the qnetworks software tool, which is currently in widespread use. The history of the development of models and the advantages and disadvantages of their work are shown.

Keywords: performance of computing systems, queuing networks, models, application packages, software tools.

Over the past decades, many application software packages (APPs) have been developed to create models for studying the performance of computing systems (CS). However, most of these tools are limited in scope, outdated, or no longer available online.

Thus, a package for researching queues based on QN network models (RESQ) [1], which was developed in the early 90s by IBM Research, attracts attention. However, since it has some advantages and disadvantages, it was one of the first successful packages for analyzing aircraft performance indicators. Let's consider it in more detail.

During its creation, in addition to the essential functions of modeling the performance of distributed networks, it also provided a modeling language for describing extended QN models. Such QN models were solved using analytical calculations or directly using modeling methods. Further, in this package, the graphical user interface (RESQME) was finalized to modify the process of determining the input parameters of QN models [2].

Further, by the 2000s, scientists finalized this package, where the QNAP2 [3] tool was created, which provided various additional methods for solving problems of modeling distributed systems. Such decisions were based on analytical calculations and simulation tools to analyze QN models.

Unfortunately, QNAP2 and RESQ are no longer available on public Internet repositories. The tools still available and in use are SHARPE, PDQ, and JMT. Among these packages, the Symbolic Hierarchical Automated Reliability and Performance Evaluation (SHARPE) [4] deserves special attention. SHARPE is an integrated package for describing and analyzing hierarchical stochastic models, including QN, fault trees, reliability models, etc.

In parallel with this, a somewhat different package Pretty Damn Quick (PDQ) [5] was developed, which was based on the use of some high-level languages , including Java, PHP, Perl, Python, and C. At the same time, PDQ began to implement the exact algorithm mean value analysis (MVA) for closed QN models.

Such a Java high-level language (JMT) modeling tool [6] was a free and open-source tool for building and evaluating QN models. This source code was developed by the Performance Evaluation Lab of Politecnico di Milano in Italy. This tool deserves attention, as it is being actively developed at present. It is cross-platform (portable to many operating systems), written in Java, and capable of processing a large class of QN models. The package supports fixed arrays, blocking, general routing strategies, priorities, and other advanced features. Its graphical interface makes the modeling process particularly efficient for the category of inexperienced users with little to no experience in analyzing QN models. Compared to the other qnetworks package, this package provides analytical solution methods that are much faster and more accurate for some model classes. As noted earlier, this package is suitable for studying many performance

indicators of the aircraft, including mechanisms for automatically building and analyzing QN models.

The development of this package made it possible to more effectively use queuing systems and Markov chains models in research and simulation problems of the CS.

Therefore, one of the effective ways to study the performance indicators of the CS is to use the notations of queuing network models (Queueing Networks, QNs), which are used in numerous APPs.

In the general case, creating notations for queuing network models is to study the performance indicators of computing systems and networks combined into complex structures consisting of several levels (layers).

Such structures are studied as open, closed, and hybrid networks, which were quite well described in the early 70s of the last century in [7].

In this regard, QN models can be estimated either using simulation tools or using analytical and numerical methods. The advantage of simulation lies in the ability to evaluate many different system structures, including extended QN models, for which other solution methods are either not available or give only approximate results.

However, such modeling techniques can take considerable time to evaluate complex models accurately, and the computed results are presented only as confidence intervals. In addition, evaluating the same model with different parameters uses ``what if'' analysis and is computationally intensive in terms of resource consumption because it involves many simulations runs.

Along with this, there are many other methods for numerically solving QN models [8] and various APPs that implement various algorithms. At the same time, the qnetworks APP, which is designed to analyze QN models and written in GNU Octave, is quite effective.

GNU Octave [9] is an interpreted language for numerical calculations that is analogous in many of its functions to MATLAB [10], with which it is mostly compatible.

Qnetworks provides a set of functions for analyzing models with and without queues, calculating performance frontiers, evaluating single-station queuing systems, and analyzing Markov chains. This package is free and open source. Users of the qnetworks system may inspect, modify, and redistribute the code by the license contained in it.

Compared to other packages such as Java JMT [11], or RESQ [1], qnetworks is not an integrated modeling tool. Qnetworks contains a library of functions that scientists can use as building blocks for analyzing QN models. Octave's interactive environment provides a framework that allows you to analyze complex model structures while providing greater flexibility quickly. Since qnetworks provides automated batch analysis, QN models can be defined and solved programmatically in the console and a graphical environment.

The disadvantage of using qnetworks is the need for a robust understanding of how QN models work. For this reason, ordinary users may prefer a less flexible but more convenient tool such as Java JMT [11].

Thus, to work with the qnetworks APP, users can apply the following different scenarios to build QN models:

- incremental model development. The qnetworks API and GNU Octave are relatively mature platforms for rapid prototyping and iterative work with QN models. Such models can be quickly defined and analyzed using the functions provided by qnetworks. Also, the Octave programming language provides additional convenient functions for working with vectors that allow you to work with various models;

- simulation environment. In the qnetworks APP, complex computer performance studies can be performed, as models and built-in structures can also be easily defined and solved. For the modeling environment, there are developed methods for working with models with the help of which tasks are solved, and standard functions are redefined;

- study of aircraft performance indicators based on queuing networks. New algorithms for analyzing QN models can be implemented inside the qnetworks package and tested against existing research data. General user contribution (creation of own libraries) is possible to further develop the qnetworks package by developing new functions. Many of the QN model algorithms described in the literature are implemented using user functions. Openness for further development of modules allows you to implement your algorithms and program code to improve work with QN models;

- education. The qnetworks APP is suitable for introducing advanced research concepts of QN models and methods for their development. When working with the qnetworks PPP, users can get advanced visualization tools for solving specific problems for QN models, including using additional graphics capabilities provided by GNU Octave.

LITERATURE

1. C.H. Sauer, M . Reiser and E.A. MacNair, ``RESQ - A Package for Solution of Generalized Queueing Networks'', Proceedings 1977 National Computer Conference (1977) pp. 977-986.

2. Chang K.C., Gordon R.F., Loewner P.G., MacNair, E.A. The Research Queuing Package Modeling Environment (RESQM E). In: Winter Simulation Conference. (1993) pp. 294-302.

3. Veran M., Potier D. QNAP2: A portable environment for queueing systems modelling. Technical Report 314, Institut National de Recherche en Informatique et en Automatique (June 1984)

4. Sahner R., Trivedi K.S., Puliafito A. Performance and Reliability Analysis of Computer Systems: An Example-Based Appro ach Using the SHARPE Software Package. Kluwer Academic Publishers (1996

5. PDQ (Pretty Damn Quick) Software Distribution : URL: http://www.perfdynamics.com/Tools/PDQcode.html (дата звернення: 25.04.2022)

6. Bertoli, M., Casale, G., Serazzi, G. JMT: performance engineering tools for system modeling. SIGMETRICS Perfo rm. Eval. Rev. 36(4) (2009) pp. 10-15.

7. F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. J. ACM, 22(2). P. 248-260, 1975.

8. Bolch G., Greiner S., de Meer H., Trivedi K. ``Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications''. Wiley (1998)

9. J. W. Eaton, D. Bateman, S. Hauberg, R. Wehbring ``GNU Octave'' : URL: https://octave.org/octave.pdf (дата звернення: 25.04.2022)

10. MATLAB. The Language of Technical Computing. : URL: https://www.mathworks.com/help/matlab/ (дата звернення: 25.04.2022)

11. Bertoli, M., Casale, G., Serazzi, G. JMT: performance engineering tools for system modeling. SIGMETRICS Perform. Eval. Rev. 36(4) (2009) pp. 10-15.

Хошаба Олександр Мирославович — канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет

Khoshaba Oleksandr M. — Cand. Sc. (Eng), Assistant Professor of Department of Software, Vinnytsia National Technical University, Vinnytsia, email : Oleksandr.Khoshaba@gmail.com